The 1998 BBN Byblos 10x Real Time System

Jason Davenport, Long Nguyen, Spyros Matsoukas,
Richard Schwartz, John Makhoul
BBN Technologies, GTE Internetworking
Cambridge, MA 02138, USA

ABSTRACT

In this paper we describe the BBN Byblos 10x real time system used for the 1998 Hub-4 English tests. Given our state of the art primary system [1] running at 230 times real time (230 xRT) we show that eliminating and approximating many computationally expensive components speeds up the system by a factor of 23 with a relative loss in WER of 18%. This is accomplished without retraining or changing the primary system structure. The components of the primary system that are refined include segmentation, adaptation, decoding, cross-word rescoring with adaptation, and system combination. The time saving algorithms used include fast Gaussian computation, grammar spreading, nbest tree rescoring, and block diagonal adaptation.

1. INTRODUCTION

Large vocabulary continuous speech recognition requires a considerable amount of computation. The amount of computation depends to a large degree on the quality of speech, with the computation increasing by a significant factor for more natural speech. Research systems frequently use 200 to 500 times real time to achieve the highest possible accuracy. While we can always decrease the computation by using an aggressive beam search pruning strategy [2], our goal here is to decode the speech with the least amount of computation while still obtaining accuracy close to that of our best research system.

In 1997 the sponsor introduced a 10 times real time contrast evaluation to show the tradeoffs between speed and accuracy on the 1997 Hub-4 English test data. We spent little time preparing for this test due to time constraints, and ultimately submitted a system that simply used smaller models, fewer parameters, and tighter pruning than our primary system. Our findings for this evaluation were based on the high quality (f0 & f1) data and showed a factor of 20 gain in speed and a relative loss in accuracy of 35%.

This year, for the 1998 Hub-4 English test, we were once again evaluated on a 10 times real time spoke. We spent more time to work on techniques that would reduce computation but have a minimal effect on accuracy for all speaker conditions (not just f0 & f1). We show that the 10x real time system is structurally the same as our primary system using the same models, number of parameters, and grammar. The majority of the speedup is from algorithms that reduce computation.

We describe these algorithms in section 2. They include fast Gaussian computation (FGC), grammar spreading, nbest tree rescoring and block diagonal transformations for adaptation. In section 3 we compare the primary and 10x systems, then we report results in section 4. For all findings in this paper assume that the results are based on the 1997 Hub-4 Evaluation data set unless otherwise noted.

2. SPEED-UP ALGORITHMS

Here we present several computation reducing algorithms used in the BBN Byblos 1998 Hub-4 10x real time system.

2.1 Fast Gaussian Computation

Generally we try to avoid speeding up one part of the computation since it doesn't result in a large factor. However, our primary system uses a very large number of Gaussian probability densities to obtain high accuracy. Thus, the computation is dominated by Gaussian evaluations in several areas of the recognition process. In the primary system, when using a narrow beam, Gaussian distance computation makes up 80% of the segmentation decoding, 76% of the forward pass decoding, 94% of the backward pass and 73% of the adapted crossword rescoring, so it is worth spending some effort to decrease this one type of computation.

We use a simpler variation of Padmanabhan's decision tree based FGC [3] to reduce the Gaussian distance computation. We start with the means of all the Gaussians in the system and build a decision tree using binary clustering [4]. Each leaf of this tree represents a unique region of the feature space. At each leaf we store a short list of the Gaussians from each codebook that are worth considering. The short lists are made by traversing the tree with labeled training data in a manner similar to that used in [3]. The algorithm determines the likely Gaussians for a codebook to be any Gaussian that was ever used within that leaf. If any codebook within a leaf has no samples in the training data, we find the Gaussian that is closest to the mean of the leaf as the sole Gaussian for this codebook.

During decoding, for each feature vector we traverse the decision tree as we do when filling the tree. This requires only an average of 2 * depth distance calculations. Then, when we need to find the most likely Gaussians for a codebook associated with a state, we only consider the Gaussians in the short list. In the forward pass, we use phonetically-tied mixtures (PTM) with 256 Gaussians per codebook. Using FGC we reduce the average number to 37 Gaussians per codebook. In the backward pass, we use state-clustered tied-mixtures (SCTM), and we reduce the average number from 64 to 23 Gaussians. We show in Table 1 the effect of FGC on the fw and bw pass using a narrow beam.

We see that the fw pass is sped up by a factor of 3 and the bw pass by a factor of 2.5 with almost no loss in accuracy.

Model	FGC	xRT	WER	# of computed Gaussians / Cbk
Fw-PTM	No	2.3	20.7	256
Fw-PTM	Yes	0.7	20.9	37
Bw-SCTM	No	1.0	20.8	64
Bw-SCTM	Yes	0.4	20.9	23

Table 1. FGC vs. No FGC using a narrow beam.

2.2 Grammar Spreading

During a beam search we generally keep any theory active if its score is within some factor of the largest path score at that frame. The beam search is clearly not admissible, because a theory that currently scores poorly might later have a better score. The basic algorithm works well if the different theories each get their scores gradually in a time-synchronous manner or the beam is so large that potentially good theories aren't pruned out. In the primary system we set the beam wide enough so that we don't remove the best scoring global path prematurely. For the 10x system, a wide beam is too costly, so we must assure that theory scores are adjusted gradually.

A major cause of irregular score changes lies with the language model costs coming not at every frame, but rather at word transitions. These language model transitions, which can often be below 10^{-6} for the correct word, occur at different times for each theory. Furthermore, we often exponentiate the language model probabilities by two or more in order to balance them against the acoustic probabilities. Thus, a theory can have its path score decrease in one frame by $10^{-12,}$ which is comparable to the width of the beam that we use in the search.

By spreading the grammar probabilities across the internal phone transitions of a word we effectively remove these large score spikes. This allows us to narrow the beam and speed up the search. Figure 1 shows a comparison of how language model costs are applied in the beam search. Previously (Old) the entire grammar cost was applied at the word transition. Now (New) the grammar cost at a word transition is reduced by the subsequent word's unigram cost. This removed cost is then applied gradually across the subsequent word's phone transitions.

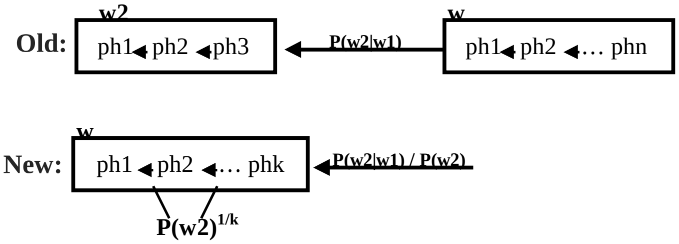

Figure 1. Spreading grammar costs across phones.

We tried trigram and bigram weighted averages for the grammar to spread, but surprisingly found that spreading the unigram probability worked best.

In Table 2 we see the effect of grammar spreading on the bw pass. It is evident that we can either reduce computation by a factor of 2 with no loss, or reduce the computation significantly for a small penalty.

Spread Grammar	Beam width	xRT	WER
No	Wide	5.2	26.3
No	Medium	1.8	29.7
Yes	Medium	2.0	27.4
Yes	Narrower	1.1	28.6

Table 2. Effect of spreading grammar in backward pass.

2.3 Nbest Tree Rescoring

In the primary system during the nbest rescoring pass we decode with crossword models each of the 100 or more nbest hypotheses separately. Typically there are only one or two words that differ in successive nbest hypotheses. For the 10x system we create a tree of these hypotheses for each reference utterance and score overlapping paths only once. This eliminates the redundancy of scoring identical partial paths of similar hypotheses. The algorithm also allows us to prune more effectively as we are scoring all theory paths in parallel. As described in the grammar spreading section, we prune based on a factor of the largest path score, so the beam width has a much larger impact on a tree of multiple hypotheses. We lose no accuracy using the nbest tree rescorer, and cut the rescoring computation by a factor of 2.

2.4 Adaptation

In the primary system we adapt the crossword DSAT [5,6] models using 2 iterations of MLLR with full matrix transformations from the results of a previous DSAT non-crossword decode. For the 10x system we use 1 iteration of MLLR using 8 block diagonal transformations using the results of the speaker independent (SI) crossword rescore. The effects of this reduced adaptation can be seen in Table 5.

Furthermore, in the primary system adapted crossword rescoring is done one speaker at a time. We use a speaker dependent (SD) acoustic model that is produced by applying the estimated transformation on the SI model prior to rescoring. This procedure is highly inefficient when there are many speakers in the test set, since a lot of time is spent in I/O and parameter initialization. In the 10x system we avoid this inefficiency by incorporating the adaptation into the rescorer. We rescore multiple speakers in one step, performing all the necessary I/O and initialization only once. The effect of adapted multiple speaker rescoring is shown in Table 3, where we see a 68% relative improvement in speed over the primary system with a minimal loss in accuracy.

	xRT	WER
Normal Adaptation	2.82	20.9
Fast Adaptation	0.88	21.1

Table 3. Effect of fast adaptation in rescoring

3. PRIMARY vs. 10x SYSTEMS

Here we compare the two systems in each recognition step.

For the 10x system, the input speech is automatically segmented as in the Primary System using a dual-band phoneme recognizer for separating channels, and a dual-gender word recognizer that locates pauses and gender changes. We speed up the word recognizer in the 10xRT system by using a combination of FGC and grammar spreading. We then cluster segments by speaker as in the Primary System to adapt the more detailed crossword DSAT models later prior to crossword rescoring. Adaptation of the non-cross-word DSAT models is eliminated.

We decode the input twice. Once with the 2-Pass N-best decoder [7] using speaker independent models as in the Primary System. And once with the adapted crossword models using the nbest tree rescorer. For the 10xRT system we eliminate non-cross-word adaptation. We also reduce the crossword adaptation to 1 iteration of MLLR using 8 block diagonal transformations compared to 2 iterations with full matrix transformations in the Primary System.

In both decoding and rescoring we use a combination of FGC, grammar spreading and pruning to reduce computation.

4. RESULTS

The recognition steps of the primary system include analysis, VTL stretch estimation, segmentation, 2-pass decode, SI cross word rescore, DSAT non cross word decode, DSAT cross word rescore and system combination. As shown in Table 3, the 10x system uses the same analysis and VTL stretch estimation, while eliminating the DSAT non-crossword decode and system combination. All other steps of the process use one or more of the algorithms described above to reduce computation and speed up the system.

We show in Table 4 the speed/accuracy tradeoff on the Hub-4 1997 evaluation data set. These results are comparable to the Hub-4 1998 evaluation results. We see that the primary system is sped up by a factor of 23 (10xRT) with a relative loss in WER of 18%. Using FGC, grammar spreading and pruning in both segmentation and decoding provides a factor of 4 savings in speed (40xRT -> 9xRT). The rest of the savings come from eliminating system combination and non-cross-word adaptive decoding along with approximating the adaptation of the crossword models.

	Primary xRT	10x xRT	WER
			14.8
Analysis	0.1	0.1	0.0
VTL Stretch Estim.	1.5	1.5	0.0
Segmentation	7.9	**0.8**	+0.5
2-Pass Decode	15.4	**4.7**	+0.7
SI xword nbest rescor	5.9	**0.9**	0.0
DSAT nonx decode	22.0	0.0	+0.9
DSAT xword rescore	14.8	**1.9**	+0.2
System Combination	163.6	0.0	+0.4
Total	**231.2**	**9.9**	**17.5**

Table 4. Speed/Accuracy for Primary vs. 10xRT systems

Acknowledgements

This work was supported in part by the Defense Advanced Research Projects Agency and monitored by Ft. Huachuca under contract No. DABT63-94-C-0063 and by the Defense Advanced Research Projects Agency and monitored by NraD under contract No. N66001-97-D-8501. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

5. REFERENCES

[1] S. Matsoukas, L. Nguyen, J. Davenport, J. Billa, F. Richardson, D. Liu, R. Schwartz, J. Makhoul, “The 1998 BBN Byblos Primary System applied to English and Spanish Broadcast News Transcription”, 1999 DARPA Broadcast News Workshop, Herndon, VA, Feb. 1999.

[2] B.T. Lowerre, “The Harpy Speech Recognition System”, PhD Thesis, Carnegie-Mellon Univ., 1976, Pittsburgh, PA.

[3] M. Padmanabhan, E. E. Jan, L. R. Bahl, M. Picheny, "Decision-tree based feature-space quantization for fast Gaussian computation", Proc. of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, CA, Dec. 1997, pp. 325-330.

[4] J. Makhoul, S. Roucos, H. Gish, “Vector Quantization in Speech Coding”, Proc. of IEEE, Vol. 73, No. 11, November 1985, pp. 1551-1588.

[5] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A Compact Model for Speaker-Adaptive Training", ICSLP Proceedings, October 3-6 1996 Philadelphia, PA.

[6] George Zavaliagkos, Jay Billa, et al. "The BBN/Byblos Hub-5 System Description", NIST 1998 Hub-5 Workshop, Linthicum, Maryland, Sep. 1998.

[7] L. Nguyen, R. Schwartz, “Efficient 2-Pass N-Best Decoder”, EuroSpeech ’97, Rhodes, Greece, Sept. 1997, pp. 167-170